Fundamenta Informaticae 96 (2009) 395–418
DOI 10.3233/FI-2009-185
IOS Press

Measuring Semantic Closeness of Ontologically Demarcated Resources

Sang Keun Rhee[*], **Jihye Lee**, **Myon-Woong Park**[*]
Intelligence and Interaction Research Center
Korea Institute of Science and Technology, Seoul, Korea
{greyrhee,myon}@kist.re.kr

Michał Szymczak, **Grzegorz Frąckowiak**, **Maria Ganzha**[†], **Marcin Paprzycki**
System Research Institute, Polish Academy of Sciences
Warsaw, Poland
{maria.ganzha,marcin.paprzycki}@ibspan.waw.pl

Abstract. In our work, an agent-based system supporting workers in an organization is centered around utilization of ontologically demarcated data. In this system, ontological matchmaking, understood as a way of establishing closeness between resources, is one of key functionalities. Specifically, it is used to autonomously provide recommendations to the user, who is represented by her/his personal agent. These recommendations specify, which among available resources are relevant / of interest to the worker. In this paper, we discuss our approach to measuring semantic closeness between ontologically demarcated information objects, while a *Duty Trip Support* application is used as a case study. General description of the algorithm is followed by a recommendation example based on support for a worker who is seeking advice in planning a duty trip.

Keywords: Ontological matchmaking, semantic closeness, recommender system, resource management, resource matching

[*]Also works: HCI and Robotics Department, University of Science and Technology, Korea
[†]Address for correspondence: IBS PAN, ul. Newelska 6, 01-447 Warsaw, Poland. Also works: Institute of Computer Science, University of Gdañsk, Gdañsk, Poland

1. Introduction

In any *knowledge space* information resources do not exist in isolation but are connected to one another through multiple direct and indirect relations, which may be explicit or inferred. Furthermore, even if two resources seem to have no relation in one environment, they may have a contextual relation within another. Existence and strength of relations of both kinds should be considered when answering the question: are two resources relevant to each other (e.g. would a given person be interested in a specific research paper?).

Today, utilization of ontologies, is considered one of more promising ways of managing heterogeneous information. In this paper, we propose an approach to measure semantic closeness among instances of ontologically demarcated resources. While the original application area was a *Duty Trip Support System*, which we have developed (for more details, see [13, 10, 5, 11]), the proposed approach can be applied in *any* situation when semantic closeness between two instances of ontologically demarcated resources has to be established / measured. Therefore, the key contribution of this paper is the matchmaking algorithm. The sample application is used to provide the motivation, the technical background and to illustrate work of the algorithm.

To this effect we start with an overview of the concept of *semantic relevance*, existing methods to establish it, and an outline of our approach. Next, we present a general scenario based on the *Duty Trip Support* application. In what follows, we utilize selected fragments of this scenario to discuss, in detail, our approach to ontological matchmaking. Material presented here extends and complements results presented in [24, 26], which should be consulted for additional details.

2. Semantic Relevance

2.1. Existing Approaches

There exist multiple approaches to establish "degree of relatedness" between information objects. One of the most widely used approaches is measuring the *semantic similarity*. It is most often utilized when dealing with lexical or text-based information, such as specific terms, or whole documents.

Measurement of semantic similarity between terms very often utilizes some structural relation. The simplest one of them is a tree-based hierarchy. Here, several approaches have been introduced including an edge-based method [21] which measures similarity on the basis of the number of edges between terms, or node-based methods [21, 19] which utilize information about the lowest common ancestors of two nodes. While it is often claimed that node-based methods are more accurate than the edge-based measure, hybrid methods have also been proposed [16, 18]. Such methods overlay the edge-based approach with the node-based one. All these methods are useful in evaluating semantic similarities within a tree-based hierarchy, but they exhibit limitations when dealing with more complex graph structures. This is especially visible in the case of cycles in the graph, or if connections between nodes represent different types of relations (e.g. characterized by different "strength" / "level of importance"), instead of simple generalization / specification of terms. Therefore, let us focus on more robust approaches.

For documents, the *semantic similarity* typically is understood as similarity of their contents, which can be measured, for instance, by lexical matching methods. Some well known methods used here are: (1) the VSM (vector space model; [25]), (2) the TF-IDF (term frequency-inverse document frequency; [17]), and (3) the LSA (latent semantic analysis; [8]). They provide an automatic way of organizing

documents into a semantic structure (e.g. for information retrieval). However, in the context of our work, it has to be noticed that these methods deal *specifically* with establishing closeness of *documents*. At the same time they are not capable of dealing with a question: will Prof. Frank Wang like food in the Golden Dragon restaurant in Tulsa, Oklahoma?

From a different point of view, similarity or relation between informational resources can be based on "links" that connect them. Here, in the case of hyper-linked web-pages the *PageRank* ([6]) approach is possibly the most well-known example. Similarly, in [15], links in the blogosphere have been interpreted as a graph-based representation of influences (i.e. the *influence graph*). Next, the *spreading activation technique* [7] was applied to the influence graph to discover influence chains and the most influential article(s). Again, this approach is not applicable to the case of a researcher from the National University of Singapore seeking suggestions of additional conference(s) and/or institutions to visit, during her upcoming business trip to the University of Calgary.

The semantic similarity can also be found in the case of generic non-textual resources. However, this typically requires additional description of those resources, e.g. using properties or tags. For instance, two persons can be considered to be similar if they have common properties; e.g. same job, same hobby, etc. Also, *tagging* can be used instead of defining properties of those non-textual resources, and a relevance measuring method based on tagging has been proposed in [14]. While this approach has some relevance to our work, it does not apply directly when a full-blown resource ontology (with graph that includes cycles and weights of properties), is utilized.

Summarizing, while there exists large body of work devoted to establishing semantic relevance of resources, proposed methods do not seem to be robust and flexible enough in the case of our application. Therefore, we propose an approach to measuring semantic similarity, which extends the above described ones.

2.2. Proposed Approach—Preliminary Considerations

Let us start from the observation that, in the case of relations between documents, sometimes it may be useful to look beyond the similarity of contents. Two documents can be *related* even though their content (e.g. described by key terms provided by the author, or terms most often appearing in the text) is not; or they are not "linked" directly with each other (e.g. they have different authors, publishers, etc.). For instance, consider a document describing an interface design methodology, and another one, explaining a data mining algorithm. Based on approaches presented above, these two documents would not be viewed as similar—since they concern two very different topics, have different authors and publishers; they are not likely to be connected/related in any way (except for possible weak link in a structural sense—they both concern topics in information technology). However, let us now consider them within a *context* of a research project devoted to the development of a recommender system. In this project, the user interface is being designed following the methodology described in the first document while the recommendation algorithm is implemented based on the contents of the second one. Obviously, these two documents are now indirectly related. In other words, the fact that methods described in both documents are utilized in the same project is the context information, and the two documents are related to each other in this context; although they are neither similar nor directly linked.

Note that semantic relevance can also be established for entities, which have completely different nature. For example, a person and a document are semantically related if the person is the author of that document. Furthermore, if this document concerns a topic for which there exists a research project, then

a relation between the author of the document and the project can be established, even if the person does not even know that the project exists.

Separately, take into account that different relationships between information objects have different strength. First, consider two books published by the same publisher. One about new methodology for software development, the other about semiotics of dances of tribes in West Australia. Second, envision two books devoted to avionics published by two different publishers. It seems natural to claim that book subject matters more than its publisher.

Finally, observe that relevance is not a symmetric relationship. For a student doing research on a certain topic, a well-known expert, an author of a book, a department in a university, or a company (all of them involved in that field) would be highly relevant. However, it would be a stretch to assume that the student would be relevant to such expert or company. In other words, since not all relation between information objects are symmetric, the relevance can also be different depending on the "point of view" (it has a "direction").

Summarizing, the *semantic relevance* we are exploring is based on a variety of (directed) relations that can be represented in the knowledge space. Furthermore, we are considering indirect relations provided via the context information. Finally, to utilize such relevance in an information system have developed, the degree of importance between any two resources has to be representable as a numeric value.

2.3. Semantic Knowledge Space

Taking into account what has been said thus far, it can be claimed that to measure the semantic relevance between information objects, it is necessary to construct a semantically structured knowledge space. A simple form of a knowledge space consists of a set of information that is to be provided to users, without any semantic annotations. Let us now follow the suggestion made above (that it is possible to measure semantic distance between arbitrary objects) and combine data objects and their users into a single category of *resources*. Here, links between so defined resources represent relations among them. In such knowledge space, connections between resources are already described, but this structure still does not have a full semantic meaning, as the relations are represented only via *syntactic links*. To describe the semantics of resources, the meaning of relations needs to be assigned to links, so that they become *semantic links*. In addition to semantically linked resources, context information can be added to represent richer meaning of relations between resources. Overall, in our approach the *semantic knowledge space* is a semantic structure which contains resources, relations between them, and the context information. Here, resources are related to each other through meaning-carrying relations, while the context information is the additional environmental or domain information related to them.

For effective knowledge management, structure of the knowledge space needs to be designed on the conceptual level, and then individual information objects can be represented within it. To represent semantic knowledge, ontologies provide a suitable formal structure. Based on these considerations, Figure 1 depicts birds-eye view of the structure of the semantic knowledge space. First, the *Conceptual Level* defines the structure of concepts, relations between them, and additional attributes. Within an ontology, this layer is the ontological structure specifying classes and their properties; but without instances. The *Individual Level* contains resources (depicted as the *Resource Layer*) and context information (depicted as the *Context Layer*) represented as instances of the ontology defined within the *Conceptual Level*. Content of these layers is semantically linked to each other, based on relations defined within the *Conceptual*

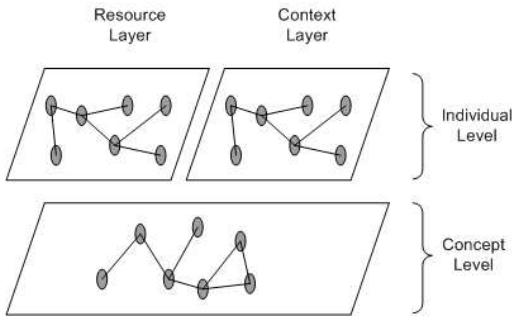

Figure 1. Knowledge Model

Level. The distinction between the *Conceptual Level* and the *Individual Level* is evident. However, the *Context Layer* and the *Resource Layer* may not be so easy to distinguish in some knowledge spaces; since some instances can be both the resource and the context information.

Note that in our work, we have made an important assumption. Across all applications, which are to work *within* a single organization, a single ontology is used. Therefore, we do not have to deal with problems related to *ontology matching/integration* (where an attempt is made to establish "common understanding" between two, or more, ontologies and/or taxonomies; see, for instance [20, 12, 9]). All that we are interested in is dealing with objects existing within a *single ontology*.

3. Duty Trip Scenario

Functional requirements and the architecture of the *Duty Trip Support* and the *Grant Announcement* applications were described in some detail in our earlier papers (see, for instance, [27, 26]). We consider *Grant Announcement* functionalities to be self explanatory and omit their discussion here. On the other hand, realization of the use case for the *Duty Trip Support* (*DTS*) is less obvious. Therefore, we intend to discuss its functionalities within a complex scenario, which covers them, and explains how the user behavior related data is stored in the semantic storage (in terms of major business objects' attributes and relations). Selected fragments of the scenario described here are going to be used to discuss our approach to semantic matchmaking. For further details, presented within a context of a slightly less complex scenario (but including, among others, ontology snippets and a detailed description of the GIS-based matching/filtering), see [26, 27]. Please note that figures found in this section do not include class definitions and *is-a* relationships; otherwise they would be hardly legible.

Figure 2 depicts the first part of our scenario (numbers refer to appropriate semantic entity instances within the figure).

- Let us consider Prof. Mai Lin, a Researcher of a *Far Eastern Research Institute* (*FERI*) who returned from a duty trip (*OldDutyTrip—(1)*) to Berlin, Germany and Kraków, Poland. Upon returning, she prepared a *Duty Trip Report* (*DTR*). The *DTR* contains important to our application, and in part required by *FERI*, information about her trip. In the context of our work we focus on the following two informations provided by Prof. Mai Lin:

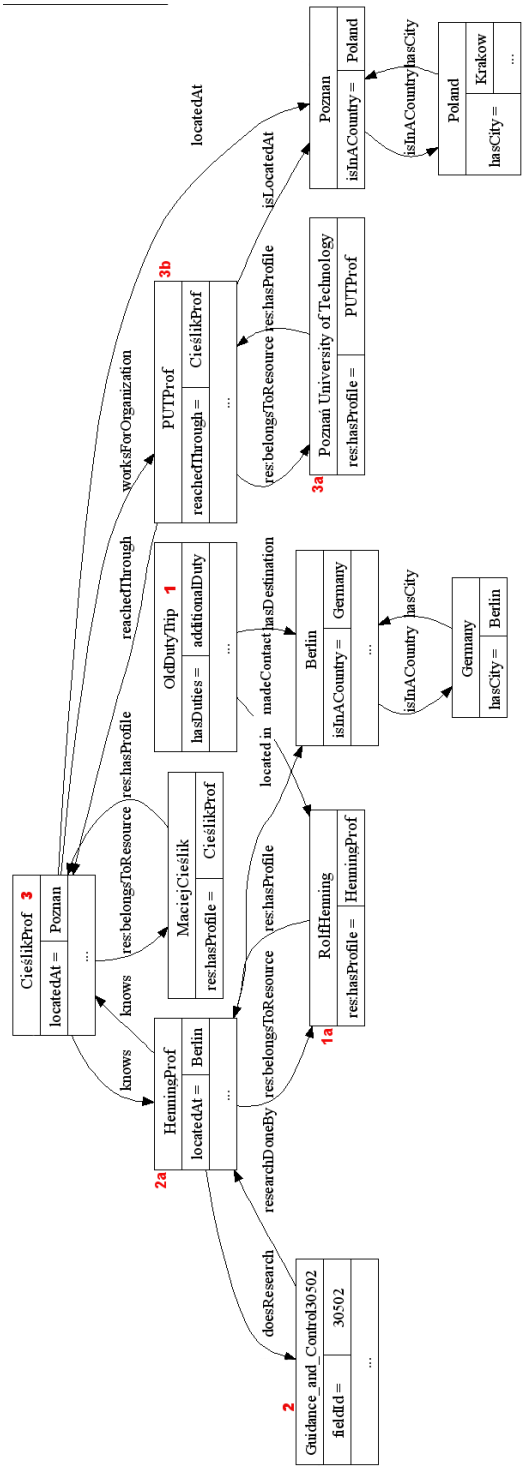

Figure 2. Past duty trip report data

1. During her trip, she met Prof. Rolf Henning (1a) who is working in the area of Guidance and Control (2), which is a sub-domain of Aerospace Ship and Ocean Engineering. This information about Prof. Henning, including fact that he works in Berlin, is stored in his profile (2a). This profile is created by the *DTS* application, on the basis of Prof. Mai Lin's *DTR*.
2. Prof. Rolf Henning knows another research scientist (from Poznań, Poland), Ph.D. Student Maciej Cieślik (3). According to Prof. Henning, Mr. Cieślik is doing interesting research in Aerodynamics (see Figure 5) and works for the Poznań University of Technology. As a result, the *DTS* application extracts and stores information about two resources located in Poznań: Mr. Cieślik, and Poznań University of Technology.

Informations about the *organization*s and the *person*s are kept in the semantic storage in the appropriate *VOResource* and *VOResourceProfile* class instances (e.g. resource representing Poznań University of Technology and its profile; (3a), (3b) respectively). In general, the *VOResource* and the *VOResourceProfile* classes represent all resources in an organization and their respective profiles (classes, and their role in the knowledge model, were discussed in detail in [10]).

- As discussed in [26], apart from activity and contact information, a *DTR* may also contain accommodation and dining recommendations. In Figure 3 we show opinions about sample objects, in Berlin and Kraków, represented in the *OldDutyTrip*). These are: restaurants GrosseSchnitzel—(4) and Wawel—(5); and hotels: SmokWawelski—(6) and BerlinPalaceHotel—(7). As far as the *DTS* application is concerned, opinions stored there originate from historical *DTR*s prepared by employees of the organization (e.g. *FERI*, in our example). Each opinion is represented in the semantic storage as an *AccommodationOpinion* or an *RestaurantOpinion* class instance (4a, 5a, 6a and 7a objects).

Now, we consider Prof. Yoon Song, researcher from the *FERI* (colleague of Prof. Lin), who is planning a duty trip to Poznań.

- Prof. Yoon Song is an Aerodynamics (8) expert and recently has been involved in an Aerospace Propulsion and Power project (9). As a result of activities involved in this project, he plans to meet, on October 7'th, 2009, with Mr. Cieślik and his colleagues, in order to discuss collaboration possibilities.
- Let us make an additional assumption: a Call for Papers for a conference (10) on Mechanical Engineering and Aerospace Ship and Ocean Engineering (10a) that is going to be held in Kraków on October 11'th, 2009 was inserted into the *FERI* database by its Administrator.
- While planning his duty trip to Poznań, Prof. Song asks the *DTS* application for recommendations of possible conference(s), people and/or institution(s) to include in his trip.
- In such a case, the conference to be held in Kraków will be recommended to him (see, Figure 4). Such recommendation is going to be based on: (a) Prof. Song's research interests (8, 9), (b) the destination of the current trip (to Poznań, which is in a close geographical proximity of Kraków), and (c) closeness of the starting date of the conference to the planned meeting in Poznań.

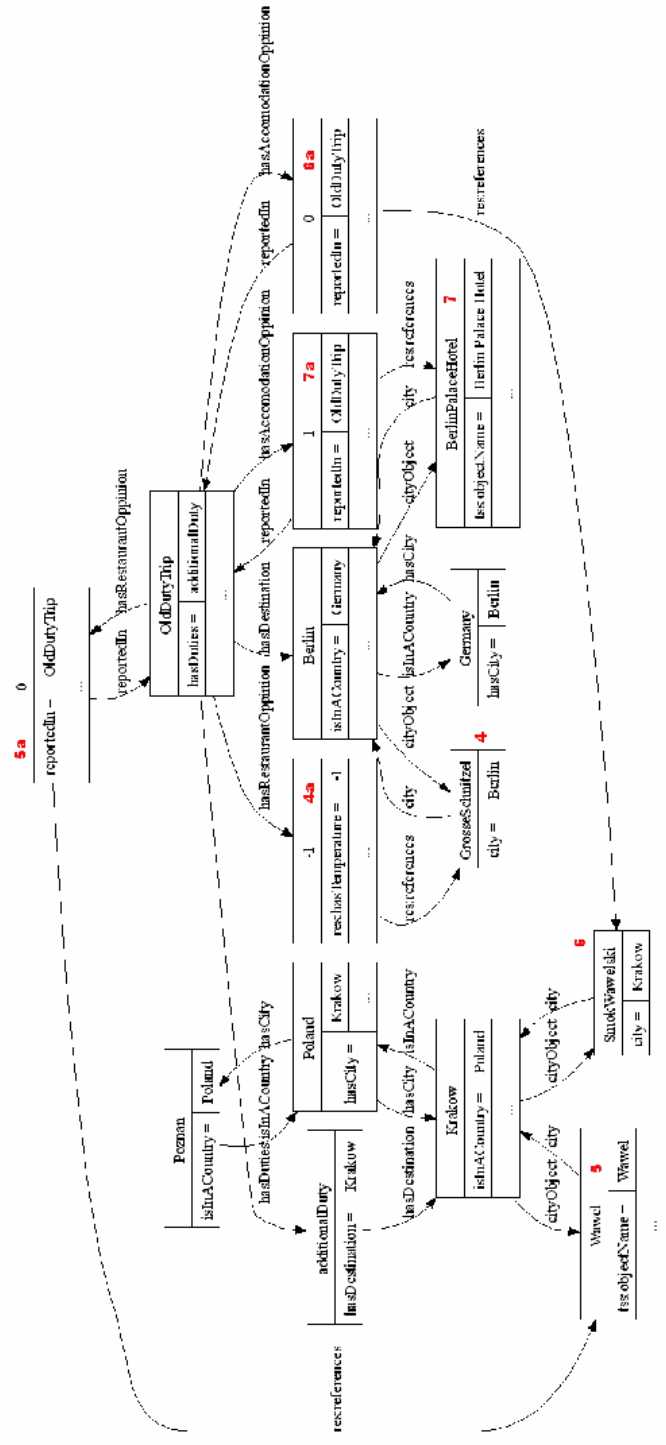

Figure 3. Travel Information Objects

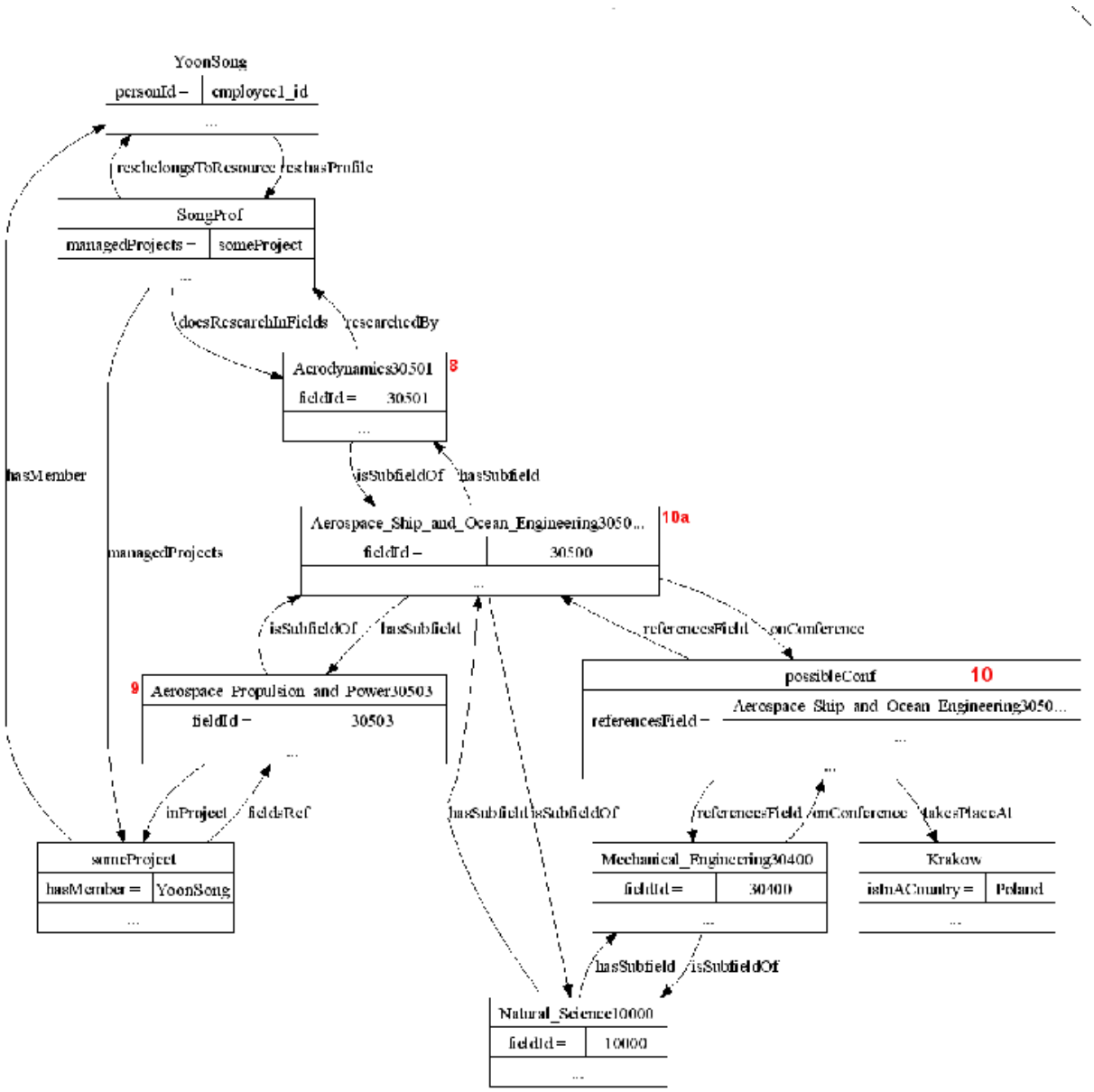

Figure 4. Kraków conference recommendation

- Similarly, Prof. Henning from Berlin ((2) in Figure 2) will be recommended as a potentially interesting contact to be visited during the trip. The relationship between the latter resource and the trip planned by Prof. Song is established on the basis of geospatial closeness of the duty trip destination (Poznań and Berlin) and similarity of research interests (recall that Prof. Henning researches Guidance and Control, which is a specialization field of Aerospace Propulsion; interest of Prof. Song; see Figure 5).

- Furthermore, as a separate advisory task, accommodation and dining recommendations can be provided if Prof. Song decides to go to the conference in Kraków (10) and/or visit Prof. Henning in Berlin. These recommendations could be: in Kraków—Wawel and SmokWawelski (5, 6), or in Berlin—GrosseSchnitzel and BerlinPalaceHotel (4, 7) (in Figure 3).

In the following sections, we will now utilize selected parts of this scenario to explain in detail the core of the system—the matching engine.

4. The Matching Engine

4.1. Matching Process Overview—Matching Criteria

Let us start by outlining the matching process that takes place within the *DTS* application (for more details, see also [26]). In the *DTS*, the main entities involved in recommendations are: *People*, *ResearchFields*, *Organizations*, *Cities* and *Countries*. However, it has to be stressed that the proposed method is not limited to these specific types of entities and works for any ontology. To find closeness (represented as a single number) between two objects we have defined the *Matching Criteria* as an ordered quadruple $\langle x, q, a, g \rangle$ (in general, the *Matching Criteria* is a tuple; see below), where:

- x is the selected ontology class instance (source object),

- q is a SPARQL query ([3]) which defines a subset of objects that are considered potentially relevant (this is the focus of the matchmaking process) and will be matched against the source object x,

- $a \geq 0$, specifies the threshold of closeness between objects to be judged actually relevant to each-other,

- in the case of our system, g is a sub-query processed by the GIS subsystem, which is responsible for finding cities which are located within a specified distance to a specified city; this sub-query is actually a triple $\langle gr, gc, ga \rangle$, where:

  - gr is an operator which allows to either limit returned number of cities of possible interest (*AMOUNT* condition) or to specify the maximum distance between the gc and the returned cities (*RADIUS* condition),
  - gc is an URI of a city demarcated with properties of the *City* class of the system ontology,
  - ga is the parameter of the gr operator ($gr(gc, ga)$); it either specifies the limit of the number of returned cities or the maximum distance between the gc and the returned cities.

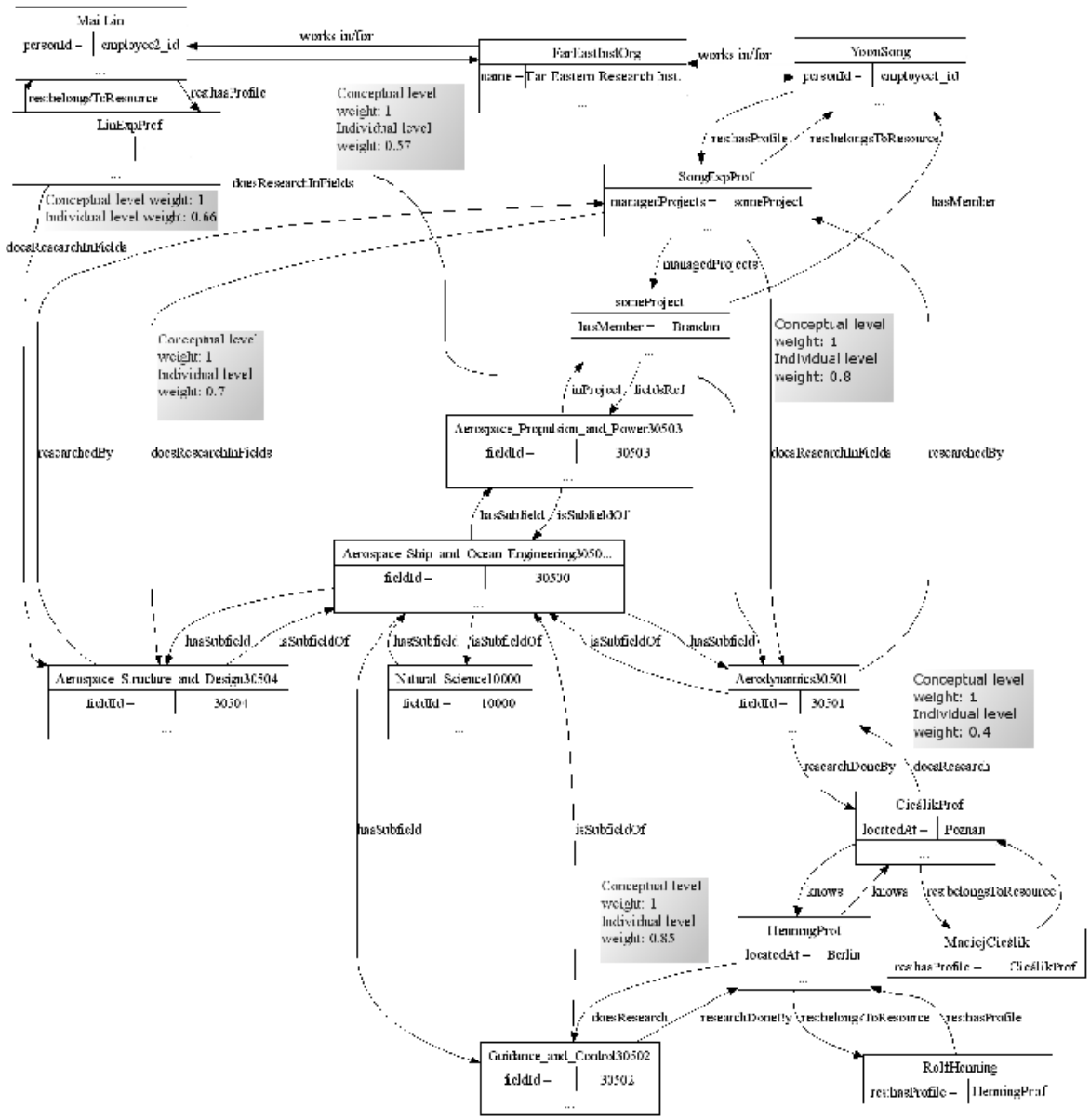

Figure 5. Research fields instances connections

In general, depending on the system and service, g is an optional query, which can be omitted, or replaced by one or more different criteria, resulting in a chain of data filtering conditions; e.g. date/time comparison, lexical matching, etc.

4.2. Core Assumptions

The *Relevance Calculation Engine* calculates closeness between instances of an ontology, based on a graph structure (the *Relevance Graph*) that represents the underlaying *Jena Ontology Model* [4]. The relevance calculation is based on the following three basic assumptions:

1. Having more relations from one object to another means that they are closer (more relevant).
2. Each relation has different importance depending on the type of the relation.
3. Even if relations are of the same type, the "weight" of the connection can vary between individual objects (instances).

The assumption 1 can be explained by a simple example. From Section 3, let us just consider three researchers—Prof. Mai Lin, Prof. Yoon Song, and Ph.D. Student Maciej Cieślik. All three of them have the same research interest, Aerodynamics. However, Prof. Lin and Prof. Song works for the same organization: *FERI*, whereas Mr. Cieślik works for the Poznań University of Technology. Therefore, it seems rather intuitive (though it may not be true in all cases) that Prof. Lin and Prof. Song are semantically closer to each other than to Mr. Cieślik (there is one more direct link between them).

The assumption 2 means that relations defined between concepts in an ontology can be weighted according to their semantic importance. Consider the same example as above. We have specified two types of relations among the three researchers: (1) having the same research interests, and (2) working for the same organization. Here, for the purpose of recommending a possible collaborator, the former relation can be regarded to be more important than the latter, as persons that work in the same organization can have completely different research interests. More specifically, from the point of view of potential collaboration the relation between *Person* and *ScientificField* is more important than the relation between *Person* and *Organization*. However, it may be the opposite if the purpose of recommendation is finding a person to proctor an exam. Therefore, it is natural to assume that different types of relations have different importance. Thus, determining their specific "weights" depends on the purpose of the system (see Section 4.3.2).

Finally, regarding the assumption 3, let us observe that some instances connected by a specific type of a relation (i.e. having the same *ontological importance weight*) may not have the same *importance* on the level of individuals. Following the same example, Prof. Yoon Song and Ph.D. Student Maciej Cieślik may both have interests in the same research topic, e.g. Aerodynamics. However, recall that Prof. Song is an expert in that area, whereas Mr. Cieślik is currently studying it. In this case, we can say that the connection between Prof. Song and Aerodynamics is stronger than that of Mr. Cieślik's. Therefore, although both Prof. Song and Mr. Cieślik are connected to Aerodynamics via the same type of relation which holds the same importance value on the *conceptual level* (according to our Assumption 2), they can have different connection "weights" on the *individual level*.

4.3. Relevance Graph

To discover the semantic relevance of ontologically demarcated resources, we first interpret the ontology as a graph structure, where the relevance can be numerically measured. A *Relevance Graph* is derived from the semantic knowledge model representing the information objects and their relations, and is a directed labeled graph $G = (V, E)$ where:

- V is a set of nodes, representing individuals,
- E is a set of edges, representing relations between individuals.

Note that the *Relevance Graph* does not have any restrictions in its structure. Here, different edges can represent different relation types, there may be multiple edges between adjacent nodes, and the graph can contain cycles. Also, depending on the specific implementation and its purpose, a whole ontology can be interpreted as a graph, or only a part of it can become the *Relevance Graph*. Let us look into this issue in more detail.

4.3.1. Node Generation

There are two ways of generating nodes in the *Relevance Graph*. First, all instances of an ontology can become nodes or, second, only instances representing specific *resources* can be interpreted as nodes. The former is simpler, but the resulting graph is larger, and thus requires more time during its generation phase and, in particular, during the relevance calculation (note that relevance calculations are one of the core operations in our system and will be repeated constantly be various entities within it). The latter requires defining additional rules to clarify the relations between resources, via context information, since those relations may disappear upon graph generation (if individuals representing context information are not included in the graph).

For example, upon generating nodes based on the ontology described in Section 3 above, every instance in the ontology can be interpreted as a node—which is the approach we take in our system, as we regard all data objects and their users as resources (see Subsection 2.3). However, depending on the system design and purpose, it might be possible to regard only some entities within the ontology as resources, which are to be transformed into nodes. Suppose that, upon node generation, we define *Persons* as resources but leave *Research_Fields* out. In this case, it is necessary to define some additional rules to describe indirect relations between *Persons* (to avoid information loss). For instance, two persons have the same research interests if they both are working on the same research fields:

$$doesResearchInFields(?r1, ?t1) \wedge doesResearchInFields(?r2, ?t1) \Rightarrow hasSameInterests(?r1, ?r2)$$

However, finding all meaningful indirect relations between resources and manually defining them as rules may not be a trivial task in itself. Hence, in practice, the recommendable approach is to adapt the method of node generation to the characteristics of a given application; selecting specific resources and context instances to become nodes, and adding only a limited number of necessary rules.

4.3.2. Edge Generation

The next step in the proposed approach is to generate edges between nodes in the *Relevance Graph*. These edges represent relations between nodes, and they are generated from relations defined in the ontology (modeled with the OWL Object Properties [2]). Included are not only explicit relations, but also relations inferred by additional rules (particularly, in the case of our system, by the OWL-DL rule set [1]). Depending on the knowledge structure, and the application, all relations existing in the ontology

can become edges or only these relations, which represent "meaningful relevance" in the system. Each edge representation is defined as follows:

- $e \in E = (x, y, d, w)$, where:
  - $x \in V$ is the *tail* node of the edge e
  - $y \in V$ is the *head* node of the edge e
  - $d \in \mathbb{N}$ is the conceptual level *distance* value (from Assumption 2)
  - $0 \leq w \leq 1 \in \mathbb{R}$ is the individual level *weight* value (from Assumption 3)

The *distance* and *weight* values will be discussed in the following sections. Although this is the full definition of an edge (in our representation) in what follows, for simplicity, we denote it $e(x, y)$ instead of $e(x, y, d, w)$.

Note that reflexive relations are ignored and do not become edges; since they have no significance in measuring relevance between nodes:

$$\forall x \in V : \qquad e(x, x) \notin E$$

4.3.3. Edge Labelling

The edges in the *Relevance Graph* represent various relations defined in the ontology. As argued above, each relation may have different importance in terms of representing closeness between objects within the knowledge space (see assumption 2 in Section 4.2). Therefore, different relations in the ontology can be weighted with different values (the *Relevance Value*) representing their importance within the system. Instead of assigning the *Relevance Value* to each edge, we assign the *Distance Value*, which is defined as the inverse of the *Relevance Value*.

$$Distance = \frac{1}{Relevance} \tag{1}$$

During the implementation, distance values can be included in the ontology—as it has been done in our system (see, [27, 26]), or in the *Relevance Graph* creation module (as seen in [23]). Currently, we do not have an automated way of assigning relevance values to each relation. Thus, it needs to be done manually by the ontology developer and/or a domain expert. In practice, all distances can be initialized with a single value (e.g. *Distance Value* = 1). These values can be later updated by applying the developers' domain knowledge and throughout testing and tuning of the working system. This was the approach undertaken in [23], and was followed during implementation of our system (see, [26]). However, there can be a (semi-)automatic way of updating the *Distance Value*, via historical evaluation methods based on the system usage and explicit/implicit feedback. This possibility will be explored and experimented with in our future work.

4.4. Relevance Calculation

Having created the *Relevance Graph*, the *Relevance* between any two nodes in that graph can be calculated. This is done utilizing the relevance calculation algorithm, which is based on our earlier work

(see, [22]). Here, as proposed above (in Section 4.2), we distinguish two levels of "scaling" of importance of ontological relations. The first one is on the *Conceptual Level* of the semantic knowledge space, and involves relationships between concepts (Assumption 2). The second one is on the *Individual Level* of the knowledge model, and specifies importance of specific properties to an individual (Assumption 3). Taking this into account, when the semantic distance is calculated, first we have to consider the distance between concepts and, second, to scale it according to "interests" of individual resources involved in matching. As an example of such process, delivery of personalized information is depicted in Figure 6.

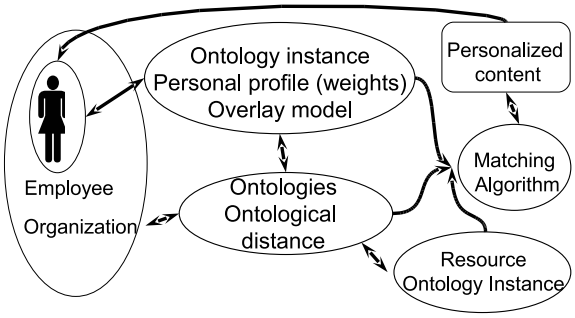

Figure 6. Top level overview of matchmaking

Here, we can see an employee within an organization. The ontology of an organization and the domain ontology provide us with a (weighted) relevance graph. At the same time, an ontological instance—the employee profile—allows us to scale specific relations in the ontology according to the employees' interests. Both the relevance graph and the individual profile, together with resources, closeness to which is to be established (selected according to the *Matching Criteria*), are the input to the matching algorithm. As an output we obtain a list of resources that are relevant to the employee. The relevance calculation process is illustrated in Figure 7. Let us now describe it in more detail.

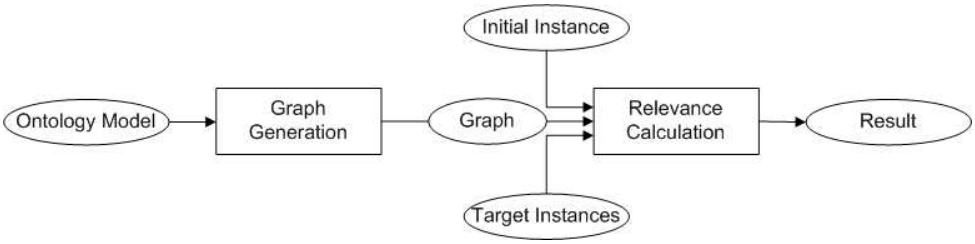

Figure 7. Relevance calculation process

4.4.1. Edge Scaling

The first step is individualizing each edge's label (i.e. distance) by applying the two importance scaling factors. The *weight* value means the degree of relevance on the individual level, represented as a real number between 0 and 1. Therefore, by multiplying this value by the *relevance value* originating from

the conceptual level, the scaled *relevance value* can be obtained.

$$newRelevance = oldRelevance \times weight \tag{2}$$

Since the *distance value* on the conceptual level resides in the *Relevance Graph*, we need to adjust the above equation taking into account equation 1. Therefore, for an edge $e(x, y) \in E$ (where x and y are the two adjacent nodes $x, y \in V$), its initial *Distance Value* $D_{e(x,y)}$, and the *weight* value w_{xy}; the scaled *Distance Value* $D'_{e(x,y)}$ is:

$$D'_{e(x,y)} = (\frac{1}{D_{e(x,y)}} \times w_{xy})^{-1} \tag{3}$$

We can regard this process as *individualization* of the edge distance. This step is an optional part in relevance calculation since it can only be applied when different *weights* are defined between instances in the ontology. Recall, that it is possible that all weights are equal to the same value.

4.4.2. Edge Merging

In the *Relevance Graph*, there may be multiple edges between two adjacent nodes, with the same direction (since two resources can be connected via multiple different relations). From Assumption 2 in Section 4.2, instead of taking the shortest (or the longest) edge, or their mean value, we apply an edge merging algorithm to obtain a simpler graph where there is only a single edge with a given direction between any two adjacent nodes. To calculate the *Distance Value* of the merged edge, we first calculate the relevance between the two nodes. For two adjacent nodes $x, y \in V$, and edges $e_1(x, y), e_2(x, y), \ldots, e_n(x, y) \in E$, linking node x with node y; we create a merged edge $e'(x, y)$. Assuming that for each edge $e_i(x, y)$, where $1 \leq i \leq n$, the distance value is $D_{e_i(x,y)}$; the semantic relevance value r_{xy} of node x to node y is as follows:

$$r_{xy} = \sum_{i=1}^{n} \frac{1}{D_{e_i(x,y)}} \tag{4}$$

Therefore, from the equation 1, the new *Distance Value* ($D_{e'(x,y)}$), of the merged edge $e'(x, y)$ is:

$$D_{e'(x,y)} = \frac{1}{r_{xy}} = (\sum_{i=1}^{n} \frac{1}{D_{e_i(x,y)}})^{-1} \tag{5}$$

This approach allows us to obtain a simpler *Relevance Graph* with cumulative strength of connections preserved in its edges. It should, however, become clear why relevance calculations for an entire data model represented in the *Relevance Graph* may be resource consuming (especially in the pre-processing phases of the algorithm). Obviously, just calculated values represent semantic relevance of adjacent nodes.

4.4.3. Path Distance Calculation

Now we need to consider the relevance between non-adjacent nodes. For two non-adjacent nodes, there may or may not exist a valid path. In our algorithm, a path is valid if and only if it contains no repeated

nodes (i.e. simple path). If a path between two non-adjacent nodes does not exist, it means that they are not related and we can consider their relevance value to be 0. If there exist one or more paths, first, we need to calculate the *Distance Value* of each of them. In graph theory, the weight (or distance) of a path in a weighted graph is the sum of the weights of edges in the path. Following this, for a path $P(a_1 a_n)$ that visits n nodes $a_1, a_2, \ldots, a_n \in V$, the path distance $D_{P(a_1 a_n)}$ is:

$$D_{P(a_1 a_n)} = \sum_{k=1}^{n-1} D_{e'(a_k a_{k+1})} \tag{6}$$

However, we have to consider the fact that there may be multiple path between two nodes. In this case, instead of selecting a single path, we consider all relations (path) existing between nodes. Unfortunately, this can produce (undesirably) close relevance of nodes connected via multiple long paths. Precisely, for a given node x, we can see that node y should be closer than node z (x and y are directly connected via a medium-strength link). However, the result could be the opposite, because of a large number of very long path from x to z. It should be obvious here, that (a) each long path indicates a relatively weak/indirect/implicit relation, and (b) that having many such links should not easily overcome strength of a direct link. Hence, it becomes necessary to adjust the *Distance Value* between indirectly related nodes; so that such connections are weaker than in the case of a simple sum. This is achieved by multiplying the *Distance Value* by the edge count k of each edge. Therefore, the above equation (6), is replaced with the following one:

$$D_{P(a_1 a_n)} = \sum_{k=1}^{n-1} (k \times D_{e'(a_k a_{k+1})}). \tag{7}$$

4.4.4. Path Merging

The last step of the relevance calculation is combining influence of multiple paths between nodes. This is done by following the same principle as in the case of edge merging. For n paths $P_1, P_2, \ldots, P_n$ from node $x \in V$ to node $y \in V$, the relevance value R_{xy} is:

$$R_{xy} = \sum_{k=1}^{n} \frac{1}{D_{P_k(xy)}} \tag{8}$$

Now, R_{xy} represents the final semantic relevance value of node y as related to node x. In summary, in Figure 8 we present the block schema of just described algorithm.

5. Matching in the *Duty Trip Support* Application

Let us now briefly illustrate utilization of the matching algorithm in the *Duty Trip Support* application. We will consider a recommendation use case described in Section 3; where Prof. Song inquires about potentially interesting person(s) to visit during his duty trip to Poland to meet with Mr. Maciej Cieślik. Note that we only briefly outline the process (with focus on ontological matchmaking), while additional information concerning each step of the calculation, with extra focus on the GIS algorithm can be found in [26].

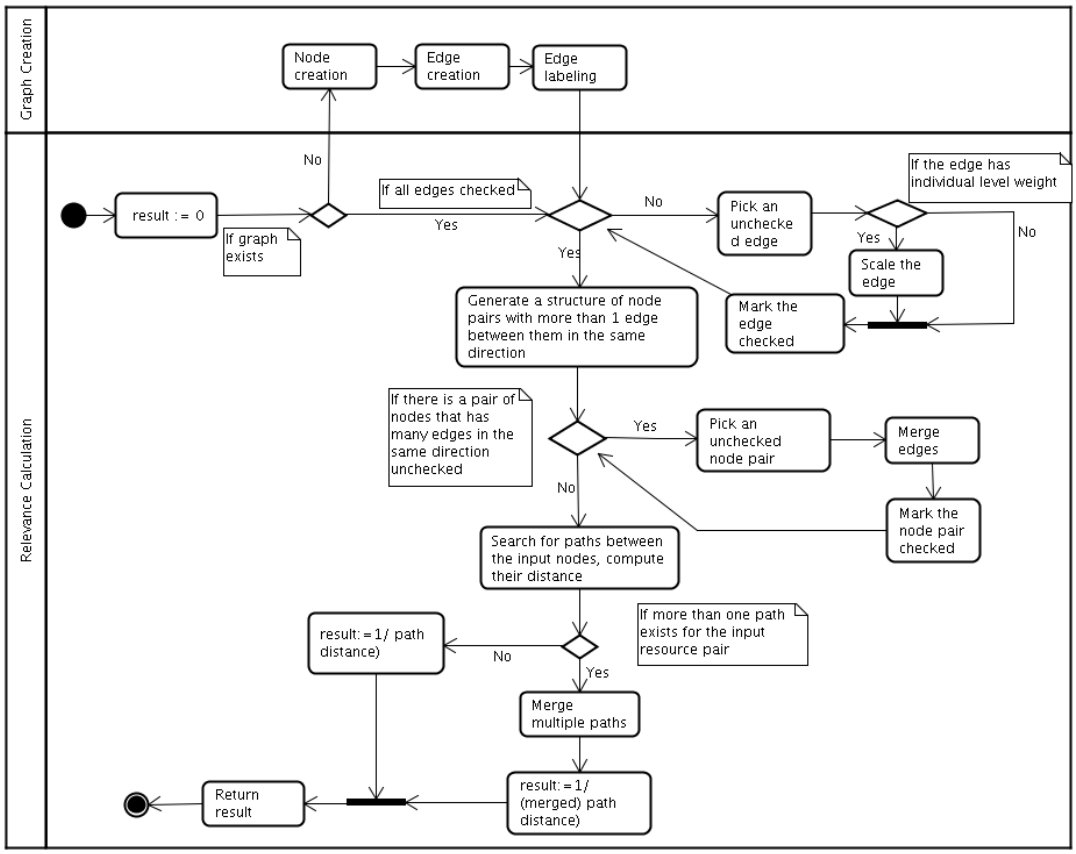

Figure 8. Matchmaking algorithm; block schema

5.1. Matching Process Example

In order to fulfill Prof. Song's request, the *DTS* application performs the following steps:

1. A *Matching Criteria* $\langle x, q, a, g \rangle$ is created such that the system can find persons potentially interesting for Prof. Song:

   (a) $x = YoonSong$

   (b) $q =$

```
PREFIX onto:
<http://rossini.ibspan.waw.pl/Ontologies/KIST/KISTVO>
SELECT ?person
WHERE {?person isa onto:ContactPerson.}
 FILTER (onto:locatedAt temp:gisResults-multi).
```

   (c) $a = \frac{1}{40}$

(d) $g = [gc, gr, ga]$:

$$gc = \texttt{Poznań},$$
$$gr = \texttt{RADIUS},$$
$$ga = 500.$$

2. By executing the GIS query g, a list of cities which are mentioned in the semantic storage at the *FERI* (that are associated with any person, organization or event) and are located no further than 500 kilometers from Poznań is returned. Recall that any past *Duty Trip Report* may result in a person or institution being added to *FERI*'s *DTS* database. Each of these individuals will have its own profile. This profile should include, among others, geospatial information. If it does, then such entity will be taken into account during geospatial processing. In the example considered here, both Kraków and Berlin would be included in the list (since they are located less than 500 km from Poznań).

3. A SPARQL ([3]) query q is formed, on the basis of the selected *Matching Criteria*, and executed. As its input, it utilizes the results from the GIS subsystem (only entities associated with cities selected by the GIS query are considered). Since in our example persons are searched, the SPARQL search engine filters out URIs of resources of the type *ContactPerson*, which describe person(s) residing in cities returned by the GIS query g. In the considered example, Prof. Henning would be selected.

4. Lastly, the *Relevance Calculation Engine* established (on the basis of semantic relevance) whether any person selected in the previous step is likely to be of interest for Prof. Song. In our example, Prof. Henning would be the target candidate. Thus, the matching process involves:

   (a) source instance $URI = YoonSong$

   (b) target objects $URI's = [RolfHenning]$

   (c) relevance threshold: $R = \frac{1}{40}$.

   If the relevance value for the object *RolfHenning* is higher than the threshold value (here, a sample value $R = \frac{1}{40}$), Prof. Henning is recommended to Prof. Song as a potential person to visit.

Let us now look into details of the last step of the algorithm.

5.2. Relevance Calculation Example

In Figure 5 in Section 3, we have presented an overview of relations between Prof. Song and Prof. Henning. However, illustrating the calculation process with all these relations would be too complex to be explanatory, hence the calculation example in this section will be based on a simpler representation, depicted in Figure 9.

Based on this figure, we can define three paths from Prof. Song to the "GIS-selected" contact person, Prof. Henning.

Path 1: *YoonSong* → *SongExpProf* → *Aerodynamics* → *CieślikProf* → *HenningProf* → *RolfHenning*

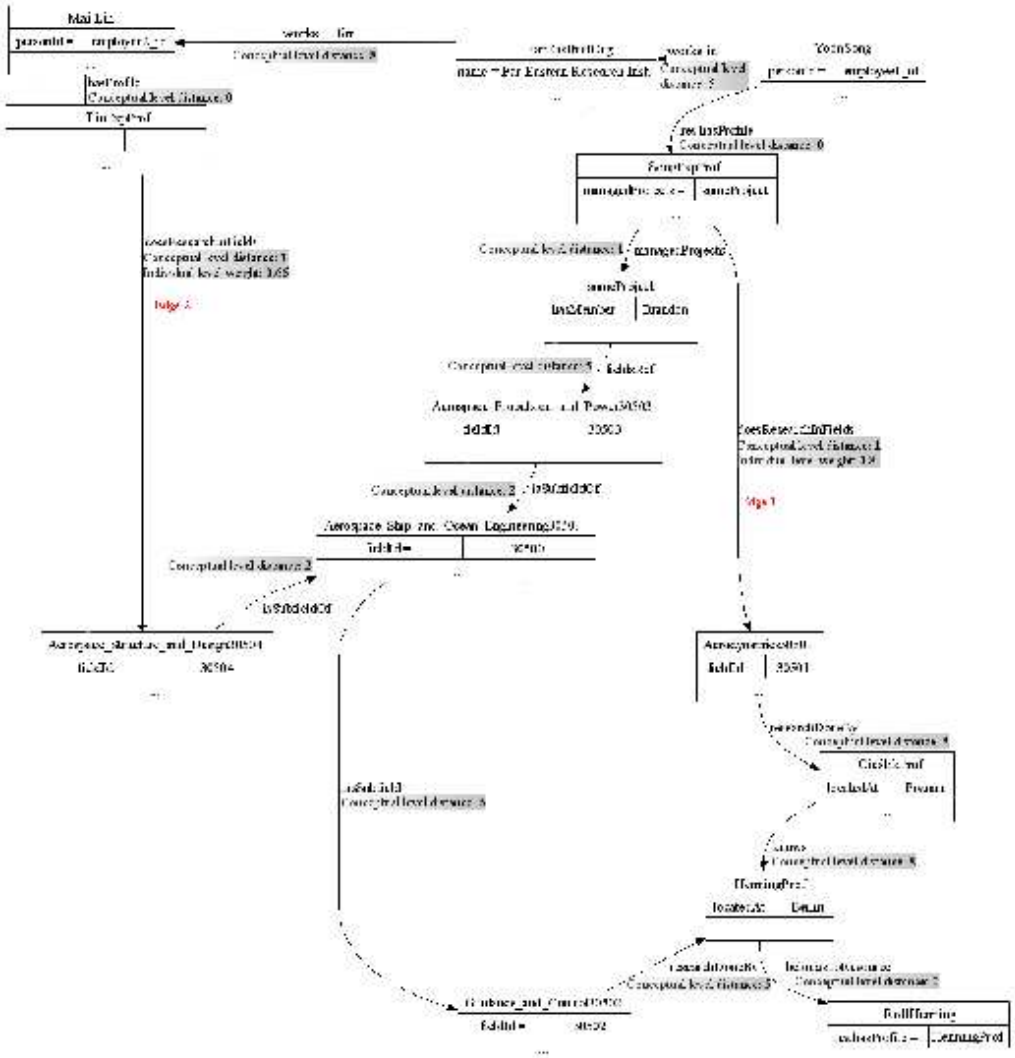

Figure 9. Simple Illustration of Research Fields and Foreign Contacts

Path 2: *YoonSong* → *SongExpProf* → *someProject* → *Aerospace_Propulsion_and_Power* → *Aerospace_Ship_and_Ocean_Engineering* → *Guidance_and_Control* → *HenningProf* → *RolfHenning*

Path 3: *YoonSong* → *FarEastInstOrg* → *MaiLin* → *LinExpProf* → *Aerospace_Structure_and_Design* → *Aerospace_Ship_and_Ocean_Engineering* → *Guidance_and_Control* → *HenningProf* → *RolfHenning*

Now, let us recall the fact that a person is extremely likely to have different levels of knowledge of individual research fields. Therefore, we first need to scale the edges of the *Relevance graph* by these individual *weights*. In Figure 9, there are only two edges(Edge1 and Edge2) that have individual weight values, and hence need to be scaled. By applying equation (3) in Section 4.4, we obtain the following individualized edge distance values:

$$D'_{Edge1} = (1 \times 0.8)^{-1} = 1.25$$
$$D'_{Edge2} = (1 \times 0.66)^{-1} = 1.515$$

Since there are no multiple edges between any two adjacent nodes, we can skip the *Edge Merging* step and proceed to calculate the distance value of each path. By applying equation (7), the respective distance values are:

$$D_{Path1} = (1 \times 0) + (2 \times 1.25) + (3 \times 5) + (4 \times 8) + (5 \times 0) = 49.5$$
$$D_{Path2} = (1 \times 0) + (2 \times 1) + (3 \times 5) + (4 \times 6) + (5 \times 2) + (6 \times 5) + (7 \times 0) = 81$$
$$D_{Path3} = (1 \times 5) + (2 \times 8) + (3 \times 0) + (4 \times 1.515) + (5 \times 6) + (6 \times 2) + (7 \times 5) + (8 \times 0) = 104.060$$

Next, the final relevance value can be obtained by merging these paths. Utilizing equation (8), we obtain:

$$Rel_{RolfHenning} = \frac{1}{49.5} + \frac{1}{81} + \frac{1}{104.060} = 0.042$$

Based on this result and the proposed *Matching Criteria*=$\{R \geq \frac{1}{40}\}$, where R is the relevance threshold, Prof. Henning will be recommended to Prof. Song. In this example, we can see that a contact person can be recommended even though (s)he does not (directly) share the same research interest. Note also, that none of the path on its own would lead to this recommendation. Only having them combined will result in the desired effect. This is important, for instance, in the case of matching of researchers with multidisciplinary research interests (where individual interests would not suffice, but their combination results would in desired matching).

5.3. Additional *Duty Trip* Construction

Similarly, additional recommendations on organizations and/or conferences can be inquired about, as a possible additional activities during Prof. Song's duty trip to Poland. All steps of the algorithm are in

principle the same as the ones described above. However, the resources sought by the SPARQL would, for instance, filter URI's of object of type *OrganizationContact* and/or *ConferenceInfo*. The relevance values would be computed based on different links between individuals in the semantic storage (links for the additional conference are illustrated in Figure 4).

Overall, as a result of the matching, for Prof. Song whose original trip is to come to Poznań to visit Mr. Cieślik, the *DTS* application would recommend (i) Prof. Henning's link as an additional contact person; (ii) the conference in Kraków as an additional conference to attend; and (iii) Poznań University of Technology as an organization to visit. This latter one may seem a bit strange, since Mr. Cieślik works at this institution, but since it has its own profile and is associated with Mr. Cieślik, it would have been correctly selected. Obviously, this special case can be easily avoided through a small modification of the algorithm. Overall, the following additional duty trip activities could be suggested:

```
:AdditionalDuty\#1 a onto:ISTDuty;
        onto:destination geo:Berlin;
        onto:personalContact :RolfHenning.
:AdditionalDuty\#2 a onto:ISTDuty;
        onto:destination geo:Krakow;
        onto:conference :KrakowConf.
:AdditionalDuty\#3 a onto:ISTDuty;
        onto:destination geo:Poznan;
        onto:orgContact :PoznanUnivOfTech.
:DTRProfile\#1 onto:duty :AdditionalDuty\#1.
:DTRProfile\#1 onto:duty :AdditionalDuty\#2.
:DTRProfile\#1 onto:duty :AdditionalDuty\#3.
```

6. Concluding Remarks

The matching algorithm described above has been implemented and is fully functional. To develop and manage our knowledge space, we utilize OWL [2] as the ontology language, Jena [4] for ontology modeling and query handling, and SPARQL [3] as the query language for retrieving and updating the resources. As a matter of fact, data presented above, for the *DTS* examples, has been prepared on the basis of the actual actions performed by the *DTS* application. First, we have stored instances / individuals discussed above; and next, obtained reported results. This and other extensive tests show that the proposed algorithm works correctly in the *DTS* application. Furthermore, for these artificially generated examples we were able to observe that resources were recommended (or not) in an intuitively correct way.

The next step is going to be running and tunning the application. This will require populating the semantic storage with a substantial amount of *actual* data. Lack of such data is the main reason that no experimental results have been reported. Obviously, we could populate the database with actual cities, actual institutions, fictitious or actual individuals that work for these institutions, and their pseudo-real research interests. However, utilization of such artificial data would make any results obtained in the process practically meaningless. While we do know that the matching algorithm works correctly, the only way to establish its utility is through extensive tests and system tuning (e.g. of the threshold values,

and weights on both the level of the model and the individual) involving actual users. This however, is clearly out of scope of this contribution. However, we are in the final stages of deploying our system in an actual *FERI* and as soon as it is deployed, we will start collecting *DTS* reports; thus populating the *DTS* semantic storage with actual data. As soon as a reasonable amount of data is collected we will be able to fine-tune the *DTS* application and report actual experimental results. These results will be presented in subsequent publications.

7. Acknowledgments

This work is partially sponsored by the KIST-SRI PAS "Agent Technology for Adaptive Information Provisioning" grant.

References

[1] OWL Web Ontology Language Guide, `http://www.w3.org/TR/2004/REC-owl-guide-20040210/#OwlVarieties`.

[2] OWL Web Ontology Language Overview, `http://www.w3.org/TR/owl-features/`.

[3] SPARQL Query Language for RDF, `http://www.w3.org/TR/rdf-sparql-query`.

[4] Jena—A Semantic Framework for Java, `http://jena.sourceforge.net`, 2008.

[5] Badica, C., Popescu, E., Frackowiak, G., Ganzha, M., Paprzycki, M., Szymczak, M., Park, M.-W.: On Human Resource Adaptability in an Agent-Based Virtual Organization, *New Challenges in Applied Intelligence Technologies* (R. K. N.T. Nguyen, Ed.), 134, Springer, Heidelberg, Germany, 2008.

[6] Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine, *Proceedings of the 7th WWW Conference*, Brisbane, Australia, 1998.

[7] Crestani, F.: Application of Spreading Activation Techniques in Information Retrieval, *Artificial Intelligence Review*, 1997, 453–482, ISSN 0269-2821.

[8] Dumais, S., Furnas, G., Landauer, T., Deerwester, S., Harshman, K.: Using latent semantic analysis to improve access to textual information, *Proceedings of the CHI'88 Conference*, ACM Press, 1988.

[9] Euzenat, J., Shvaiko, P.: *Ontology Matching*, Studies in Computational Intelligence, Springer, Heidelberg, Germany, 2007.

[10] Frackowiak, G., Ganzha, M., Gawinecki, M., Paprzycki, M., Szymczak, M., Park, M.-W., Han, Y.-S.: *Considering Resource Management in Agent-Based Virtual Organization*, Studies in Computational Intelligence, Springer, Heidelberg, Germany, 2008, In press.

[11] Frackowiak, G., Ganzha, M., Gawinecki, M., Paprzycki, M., Szymczak, M., Park, M.-W., Han, Y.-S.: On resource profiling and matching in an agent-based virtual organization, *Proceedings of the ICAISC'2008 conference*, LNCS, Springer, 2008.

[12] Gangemi, A., Pisanelli, D. M., Steve, G.: Ontology Integration: Experiences with Medical Terminologies, *Proceedings of Formal Ontology in Information Systems, FOIS'98* (N. Guarino, Ed.), IOS Press, 1998.

[13] Ganzha, M., Paprzycki, M., Gawinecki, M., Szymczak, M., Frackowiak, G., Badica, C., Popescu, E., Park, M.-W.: Adaptive Information Provisioning in an Agent-Based Virtual Organization—Preliminary Considerations, *Proceedings of the SYNASC Conference* (N. Nguyen, Ed.), 4953, IEEE Press, Los Alamitos, CA, 2007.

[14] H. Cho, T. I., Inaba, R., Takasaki, T., Mori, Y.: Pictogram Retrieval Based on Collective Semantics, *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, 4552, Springer, Springer Berlin / Heidelberg, 2007.

[15] Java, A., Kolari, P., Finin, T., Oates, T.: *Modeling the Spread of Influence on the Blogosphere*, Technical report, March 2006.

[16] Jiang, J., Conrath, D.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, *Proceedings of the International Conference on Research on Computational Linguistics*, Taiwan, 1997.

[17] Korfhage, R.: *Information Storage And Retrieval*, John Wiley & Sons, New-York, 1997.

[18] Leacock, C., Chodorow, M.: Combining Local Context and WordNet Similarity for Word Sense Identification, *An Electronic Lexical Database*, 1998, 265–283.

[19] Lin, D.: An Information-Theoretic Definition of Similarity, *15th International Conference on Machine Learning*, Morgan Kaufmann Inc., San Francisco, CA.

[20] Pinto, H. S., Martins, J. P.: Ontology Integration: How to perform the Process, *Proceedings of the IJCAI-01: Workshop on Ontologies and Information Sharing*, Seattle, Washington, USA, 2001.

[21] Resnic, P.: Using Information Content to Evaluate Semantic Similarity in a Texonomy, *International Joint Conferences on Artificial Intelligence*, 1995.

[22] Rhee, S. K., Lee, J., Park, M.-W.: Ontology-based Semantic Relevance Measure, *Proceedings of the SWW 2.0 (ISWC)*, 294, CEUR-WS, 2007.

[23] Rhee, S. K., Lee, J., Park, M.-W.: RIKI: A Wiki-based Knowledge Sharing System for Collaborative Research Projects, *Proceedings of the APCHI 2008 Conference*, LNCS, Springer, 2008.

[24] Rhee, S. K., Lee, J., Park, M.-W.: Semantic Relevance Measure between Resources based on a Graph Structure, *Proceedings of the IMCSIT'08 Conference*, IEEE, 2008.

[25] Salton, G., Wong, A., Yang, C. S.: A Vector Space Model for Automatic Indexing, *Commun. ACM*, **18**(11), 1975, 613–620, ISSN 0001-0782.

[26] Szymczak, M., Frackowiak, G., Ganzha, M., Paprzycki, M., Rhee, S. K., Lee, J., Sohn, Y. T., Kim, J. K., Han, Y.-S., Park, M.-W.: Ontological Matchmaking in an Duty Trip Support Application in a Virtual Organization, *Proceedings of the IMCSIT'08 Conference*, IEEE, 2008.

[27] Szymczak, M., Frackowiak, G., Ganzha, M., Paprzycki, M., Rhee, S. K., Park, M.-W., Han, Y.-S., Sohn, Y. T., Lee, J., Kim, J. K.: Infrastructure for Ontological Resource Matching in a Virtual Organization, *Proceedings of the IDC Conference* (N. Nguyen, R. Katarzyniak, Eds.), 134, Springer, Heidelberg, Germany, 2008.